

View Sync Diff Revs Diff vs ClientFile Describe Changelist Help

OK

Revision File Name Changelist Date Description

3	//depot/m...	2357	2000/01/14	Merge in calendar sync changes from Continuous
2	//depot/m...	2265	2000/01/13	Apply 2.0 files
1	//depot/m...	2100	2000/01/10	Add redirector to P4

Actions

Display Branching History

```

5  /*****
   usercontrol.cpp - UserControl class.
   Description:
   See usercontrol.h for description.
   Original Author: Harry and Craig?
   Modified By: Rob Oliver
10  (C) 1997, 1998 Research In Motion Ltd.
   *****/

   #pragma warning(disable:4201)
   #pragma warning(disable:4514)
   #pragma warning(disable: 4786) // identifier was truncated to '255' characters in the browser information
   #pragma warning(disable: 4505) // unreferenced local function has been removed

   // disable warnings produced by STL
   #pragma warning(disable: 4100) // unreferenced formal parameter
   #pragma warning(disable: 4663) // C++ language change
   #pragma warning(disable: 4244) // 'return' : conversion from 'const int' to 'char', possible loss of data
   #pragma warning(disable: 4018) // signed/unsigned mismatch
   #pragma warning(disable: 4710) // function not expanded
   #pragma warning(disable: 4512) // assignment operator could not be generated

30  #include "usercontrol.h"

   #include <assert.h>
   #include <process.h>
   #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   #include "scs.h"
   #include "debug.h"
   #include "mailbox.h"
   #include "time.h"
   #include "ce.h"
   #include "format.h"
   #include "icalformat.h"
   #include "timer.h"
35
40

```

```

3830 ReportEx(DebugLog::LOG_WARNING, fcn, "DecryptDecompress() failed, Tag=%d", etpld);
SendTransactionErrorStatusToDevice(GME::E_DECRYPTION_ERROR, transid);
goto end;
}

    unsigned char cmd;
    if (!GetOneByte(ClearStream, cmd)) {
3835 ReportEx(DebugLog::LOG_WARNING, fcn, "Bad format, Tag=%d", etpld);
SendTransactionErrorStatusToDevice(GME::E_BAD_GME_FORMAT, transid);
goto end;
}

    switch (cmd) {
    case GME::MESSAGE_FROM_PAGER:
    case GME::MESSAGE_FROM_SERVER: // debug
3840 //ReportEx(DebugLog::LOG_DEBUG, fcn, "MESSAGE_FROM_PAGER");
if (!Istream(pszContentType, "CMIME")) {
// Mail item
retval = HandleMessageToSubmit(ClearStream, transid, etpld);
}
else if (!Istream(pszContentType, "CICAL")) {
// Calendar item
retval = HandleAppointmentToSynchronize(ClearStream, transid, etpld);
}
else {
// Unknown content type
retval = false;
}
break;
    case GME::MORE_REQUEST:
3855 //ReportEx(DebugLog::LOG_DEBUG, fcn, "MORE_REQUEST");

    int refId;
    int requestedOffset;
    int requestedLength;
    int partId;
    if (!GetIntegerData(ClearStream, refId) ||
        !GetIntegerData(ClearStream, partId) ||
        !GetIntegerData(ClearStream, requestedOffset) ||
        !GetIntegerData(ClearStream, requestedLength)) {
3865 ReportEx(DebugLog::LOG_WARNING, fcn, "Bad format, Tag=%d", etpld);
SendTransactionErrorStatusToDevice(GME::E_BAD_GME_FORMAT, transid);
}
    else {
3870

```

```

4130 bool IsRetryable = false;
4131 do
4132 {
4133     status = mailbox->Synchronize( &ApptToCalendar );
4134     } while ( (status != true) && (retryCnt++ < MAX_RETRIES_ON_MAILBOX_FAILURE) &&
4135             (mailbox->GetLastError(&IsRetryable), IsRetryable) );
4136
4137 if ( status )
4138 {
4139     ReportEx(DebugLog::LOG_DEBUG, fcn, "Calendar appointment synchronized for device, Tag=%d", etpId);
4140 }
4141 else
4142 {
4143     Mailbox::ERR_CODE mailRc = mailbox->GetLastError();
4144
4145     // ndfh8. Send back GME::message_error to device with refId from message.
4146     if (mailRc == Mailbox::ERR_NOT_FOUND) {
4147         ReportEx(DebugLog::LOG_DEBUG, fcn, "Send() failed: %s, Tag=%d", mailbox_errors[mailRc], etpId);
4148     }
4149     else {
4150         ReportEx(DebugLog::LOG_WARNING, fcn, "Send() failed: %s, Tag=%d", mailbox_errors[mailRc], etpId);
4151     }
4152
4153     switch(mailRc) {
4154
4155     default:
4156     case Mailbox::ERR_FAIL:
4157         errCode = GME::E_GENERAL_MESSAGE;
4158         break;
4159
4160     case Mailbox::ERR_RESOURCE_ALLOC:
4161         errCode = GME::E_MAILBOX_RESOURCE_FAILURE;
4162         break;
4163
4164     case Mailbox::ERR_MAIL_CREATION:
4165         errCode = GME::E_MAIL_CREATION_FAILURE;
4166         break;
4167
4168     case Mailbox::ERR_SUBMIT_MAIL:
4169         errCode = GME::E_MAIL_SUBMIT_FAILURE;
4170         break;
4171
4172     case Mailbox::ERR_BAD_RECIP:

```